11-2008

Ubi-App: A Ubiquitous Application for Universal Access from Handheld Devices

Shameem Ahmed
Marquette University

Moushumi Sharmin
Marquette University

Sheikh Iqbal Ahamed
Marquette University, sheikh.ahamed@marquette.edu

Universal Access in the Information Society, Vol. 7 (November 2008): 273-283. DOI.

Ubi-App: A Ubiquitous Application for Universal Access from Handheld Devices

Shameem Ahmed
Department of Mathematics, Statistics and Computer Science, Marquette University, Milwaukee, WI
Moushumi Sharmin
Department of Mathematics, Statistics and Computer Science, Marquette University, Milwaukee, WI
Sheikh I. Ahamed
Department of Mathematics, Statistics and Computer Science, Marquette University, Milwaukee, WI

Abstract

Universal access from a handheld device (such as a PDA, cell phone) at any time or anywhere is now a reality. Ubicomp Assistant (UA) (Sharmin et al. in Proceedings of the 21st annual ACM symposium on applied computing (ACM SAC 2006), Dijon, France, pp 1013–1017, 2006) is an integral service of MARKS (Sharmin et al. in Proceedings of the third international conference on information technology: new generations (ITNG 2006), Las Vegas, Nevada, USA, pp 306–313, 2006). It is a middleware developed for handheld devices, and has been designed to accommodate different types of users (e.g., education, healthcare, marketing, or business). This customizable service employs the ubiquitous nature of current short range, low-power wireless connectivity and

readily available, low-cost lightweight mobile devices. These devices can reach other neighbouring devices using a free short-range ad hoc network. To the best of the authors' knowledge, the UA service is the only service designed for these devices. This paper presents the details of Ubi-App, a ubiquitous application for universal access from any handheld device, which uses UA as a service. The results of a usability test and performance evaluation of the prototype show that Ubi-App is useful, easy to use, easy to install, and does not degrade the performance of the device.

Introduction

The worldwide usage of handheld mobile devices [Personal Digital Assistant (PDA), smart phone, cell phone, etc.] through 2005 was more than 500 million devices [1]. This was a substantial increase from 1999, when overall usage was around 200 million devices [18]. In almost every aspect of life, there exist potential applications for pervasive computing technology [21]. Hoping to provide more and more support for the users' personal and professional activities, many applications have been developed, such as schedulers, notepads, calculators, file explorers, pocket words, pocket excels, memo pads, tracking utilities, and Structured Query Language (SQL) queries. Other applications include games, financial software (Tip & Tax Calculator, Money Manager, Salary Calculator), travel related software (Mass transit map, Fuel Log, etc.), communication software (digital clock, address book with pictures), international software (currency converter, international country codes, foreign language dictionaries), and web-based software (Internet Explorer, e-mail, Pocket MSN) [1]. Some tools have also been developed to support application developers. Unfortunately, in order to develop any application based on a wireless ad hoc network, expertise is required in many core functionalities, including operating systems and networking. Also, the developer needs to manage a wide variety of issues (user, context, and resource sensitivity, secure communication among devices, etc.) in the code. In most cases, this is difficult if not impossible to accomplish. Hence, a need exists for a middleware that helps developers to develop and manage omnipresent applications. This was the main goal in developing the Ubicomp Assistant (UA).

Recently, middleware projects have been developed (e.g., RCSM, Gaia, Oxygen) which provide a specific service for specific users in a specific place [4, 8, 9, 14, 22]. Unfortunately, existing middleware solutions do not support any customizable ubiquitous service, which can be utilized by multiple users at any time and at any place, without the need for any additional resources. One obvious but inefficient solution is to design a particular service for specific users. In that case, both the application developers and the end users need to be aware of every service in order to deploy them correctly in different ubiquitous applications. Thus, it is obvious that having an omnipresent and adaptable service, according to the needs of the users, is critical. Moreover, the service should have an appropriate ad hoc wireless infrastructure capacity, as well as the capability of coupling the physical world to the mobile information world. The abilities to work within a heterogeneous platform and obtain free availability would also be useful. These were the main goals in developing of UA service. This paper presents the details of an application called Ubi-App, which uses UA as a major service.

Recently, the first prototype of the middleware named middleware adaptability for resource discovery, knowledge usability and self-healing (MARKS) was developed [16]. Both the core and supplementary services (e.g., Knowledge Usability, SAFE-RD, GETS Self-healing, PerAd service, Security service, etc.) are supported by MARKS middleware [16]. However, those services are tied to a specific user and context. To make MARKS truly ubiquitous, UA [17] has been incorporated within it as a component.

Section 2 presents some scenarios to describe the necessity of the UA service, as well as of Ubi-App. Related work is presented in Sect. 3. The characteristics of Ubi-App are presented in Sect. 4. The major features and functional requirements of Ubi-App are described in Sect. 5. Section 6 discusses how Ubi-App satisfies those requirements. In Sect. 7, the use of Ubi-App is illustrated. The implementation and evaluation process is reported in Sect. 8. The paper concludes with some novel directions for future research in Sect. 9.

Motivation

Despite the different physical constraints of handheld devices running in the ubiquitous computing environment, such devices incorporate many of the same rich features used by more established technology, such as desktop PCs and laptops. MARKS has been developed with these features in mind. Ubicomp Assistant can be used by various users in different situations. The subsequent subsections present several pragmatic scenarios where UA as well as Ubi-App can be effectively used.

Scenario 1

After visiting a museum, the instructor (Dr. Factor) and her students are returning to their institute by bus. Dr. Factor has prepared a questionnaire, which is comprised of new questions, as well as some previous exam questions taken from the database stored in her PDA. Dr. Factor transmits those questions to the students' PDAs in order to evaluate their understanding. After receiving the questions, each student begins answering on their devices. Since Joseph (a student) is confused about a specific question, he sends a query to the instructor and quickly receives feedback from her. Shortly thereafter, every student sends their answers to Dr. Factor. The answers are reviewed and grades are assessed automatically, and the results are distributed to the students accordingly.

Scenario 2

Dr. Ross is presenting a very exciting topic in a conference. Dr. Jackson, a conference attendee, really likes the presentation and would like an electronic copy of the presentation. However, Dr. Ross does not want to make those documents available through the Internet. Another conference attendee, Mr. Peterson, arrives late and misses some important points presented by Dr. Ross. Therefore, Mr. Peterson would also like the files. Both Dr. Jackson and Mr. Peterson start their Ubi-App on their own PDAs and request the files from Dr. Ross's PDA. Since these two people were authorized to receive conference materials upon pre-registration, Dr. Ross's Ubi-App agrees to send them the necessary documents, and requested files are transferred through the ad hoc network of PDAs.

Scenario 3

Fun lover Jim goes to a downtown coffee shop to have some coffee and to spend some time chatting with nearby people. He launches the Ubi-App in his PDA and the program finds trusted PDAs nearby. Laura and Rachel, who are in the same coffee shop, are among those willing to chat, and Jim starts chatting with them. Suddenly, Ubi-App shows that one of his buddies, Jerry, is also available within a short range (but not in the coffee shop). Without any delay, Jim starts chatting with Jerry too, without disconnecting from Laura and Rachel.

Scenario 4

In the chemistry lab, students are working in groups of two or three. Several students have questions and since they do not want to disturb others, they simply write down their questions and send them, using Ubi-App, to the lab instructor's PDA. If the lab instructor sees that many students have the same question, he can broadcast the answer to every student's PDA. For an infrequent question, he merely sends the answer back to the group who asked.

Scenario 5

Linda, a representative of XYZ Company Inc., is sitting in a market place and receives a phone call from her manager requesting that she create a survey on the "ABC" product, distribute it to interested customers, and collect their responses. Linda opens her Ubi-App, logs in as Pollster, and creates the corresponding survey. She sends the survey to her manager for review and, upon receiving her manager's changes, she finalizes the survey

questions. She then broadcasts the questions (including an enticement that every survey participant will get a 10-dollar credit from XYZ Company) using her PDA to the nearby people in the market. Many people become interested, complete the survey through Ubi-App, and send their answers to Linda. After collecting all the responses, Ubi-App automatically shows the summary (database and graph) of the responses to the survey so that Linda can show them to her manager.

Scenario 6

Dr. Jones, a resident doctor in a local hospital, has collected important information about his patients. He has also taken some notes and observations. Dr. Jones is leaving the hospital and a new doctor, Dr. Khan, will cover for him. Before signing off, Dr. Jones needs to inform Dr. Khan about the status of his patients. He distributes his notes and observations to Dr. Khan so that the new doctor does not need to waste his valuable time recollecting this information from the patients.

A human being performs different tasks in different scenarios; hence an assistant is needed which will work in each dynamic situation. Ubi-App is a complete service provider for all the above scenarios.

Related work

Recently, some research work has been conducted on smart spaces. The Oxygen project [14] introduced the concept of an “intelligent space” occupied by cameras, microphones, displays, sound output systems, radar systems, wireless networks and controls for physical entities. People can interact by using speech, hand gestures, drawings, and body movement. Gaia [8] is another project also based on active space. The approach proposed in this paper is different. The main underlying consideration is that the user may be anywhere and may want to access services at any time.

D. B. Stewart [21] mentions the need for a new software engineering method. His paper details why software should be “miniature software.” In the work reported here, the “miniature” concept has been deployed, and additionally, a service which enables the users to utilize the device almost anywhere in an environment has been developed.

In SEREFE [3], a new architecture for SEREndipitous File Exchange was presented. Using SEREFE, each user can easily share information with other users and devices, reducing the need to plan what files they will share and when they will share them. However, in this architecture, Internet connectivity is a basic requirement. On the contrary, the approach proposed in this paper does not require any Internet connectivity. The only requirements are wireless communication (IEEE 802.11 or Bluetooth) and running MARKS.

In Conference Assistant [6], the authors present a prototype for assisting conference attendees in choosing presentations to attend, taking notes, and retrieving those notes. They also discuss the important relationship between context-awareness and wearable computing. In the approach of this paper, the Conference Assistant is a part of Ubi-App although the user interface and file transfer utility are different.

ConChat [15] is a context aware chat program that improves electronic communication. To accomplish this, it provides contextual information and resolves potential semantic conflict between users. Ubi-App also provides chatting capability among users.

According to Fischer and Konomi [7], education should be a lifelong learning process which should help an individual enhances his or her abilities to learn, engage in meaningful activities, exploit the power of media, etc. They also detailed how distributed intelligence can help enhance education and learning. The examination part of the developed application adopts that perspective.

Karagiannis et al. [13] discuss a personalized homecare system where doctors, nurses, and patients are the users. Ubi-App provides a service, which would be useful to keep record and sharing the patient information among the doctors; thus, this application will ultimately accelerate patient care process of healthcare system. Moreover, doctors, nurses, and patients can also use the application in a way adapted to their own requirements.

In [10–12], Holzinger shows details of the development of user-friendly interfaces according to the User-Centered Design (UCD) method through a rapid prototyping model. In each phase of the development of Ubi-App, representatives of different end user groups were and continue to be consulted. The initial requirements for different user groups were collected from faculty members, doctors, and students. After compiling these requirements, they were revised and approved by representatives of different end user groups. The initial prototype was evaluated by participants from each of the end user groups, and the prototype was modified according to their comments.

According to Stephanidis et al. [20], universal access demands couple of basic requirements such as *accessibility* and *high quality* of interaction by anyone irrespective of time and place. These requirements have been incorporated in Ubi-App. Adaptive techniques can play a critical role to facilitate those requirements [19]. By being customizable, Ubi-App also includes that property in it.

Characteristics of Ubicomp Assistant

Ubi-App is targeted to users in different environments. The diversity of different applications means that different requirements are needed for different user roles in different contexts. To meet the major requirements of each user, Ubi-App and UA encompass the following features:

C1: Customizability

Customizability based on varying user needs is one of the main features of Ubi-App. To satisfy every user, the functionalities of Ubi-App are customizable. As the same software is used by every user, it will be necessary to provide or restrict different user roles with a different set of functionalities. For example, an instructor should have the capability of creating, editing, and distributing an exam, while the student should have restricted features such as exam taking and answer submission. Accordingly, user interfaces as well as operations need to adapt automatically.

C2: Multi-dimensional utility

The capabilities of a single application can be endless. However, with the increasing support for different functionalities, the size of the software increases. To design applications that can serve the need of varying users, in various ways, requires consideration of the user needs and capability of mobile devices to support these functionalities. Target users of Ubi-App are teachers, students, resident doctors, etc. To satisfy the needs of these users, Ubi-App provides multi-dimensional services. It is used for creating, editing, and distributing a survey or an exam on the fly. It has chatting options and file-transfer features, and also has the capability of note taking and maintaining different types of databases (exams, survey questionnaires, patients, etc.). Ubi-App can also be extended to add any other feature.

C3: Heterogeneous platform

Support for a heterogeneous platform is highly recommended for any application. Different users may work with various types of computing devices such as PCs, laptops, PDAs, or smart phones. It is a desirable feature that an application designed for laptop can also run on PDAs without much effort or modifications. Ubi-App can be executed in any of these devices, provided the devices have a wireless communication feature.

C4: Omnipresent seamless service capability

The notion of ubiquitous computing is based on the idea that users should enjoy computing support anywhere and anytime. Thus, Ubi-App has the capability to be used seamlessly at any place and at any time. Any inconvenience (refusing the user access to the network, partial file transfer due to an unknown reason, etc.) may undermine the user's confidence in the entire process.

C5: Ad hoc wireless infrastructure expertise

Ubi-App does not require a fixed infrastructure, although having a fixed infrastructure should not impede its operation. Since any user can join or leave a group at any time, Ubi-App should have the expertise to utilize an ad hoc wireless infrastructure.

C6: Miniature footprint

According to [21], miniature software is needed for pervasive computing technology. Since presumably Ubi-App would be used mainly in tiny devices that do not have a large memory space, a small memory footprint is a primary requirement for this service.

C7: Free availability

To get services like Internet capability, cellular phone service, etc. in mobile devices, the user needs to subscribe to a service provider at a nominal cost. However, the service of Ubi-App should be free. Thus, Ubi-App will be freely available to expedite communications ubiquitously.

C8: Persistent connectivity

The connection between two devices should be persistent so that the user feels that he/she is "ON" while his/her device and wireless connection are active.

C9: Capability of coupling the physical world to the mobile information world

Ubi-App provides a way for the user to transparently couple the physical world to the mobile information world. For example, in the physical world, a fun lover may want to find their buddy nearby to have a friendly chat.

Functional requirements of Ubi-App

Ubi-App is user specific, i.e., the characteristics, user interface, and features should automatically be updated according to the user's role (instructor, student, pollster, survey taker, conference attendee, note taker, etc.). However, the role of a user is time and place dependent. That means that one person can be a student in an exam at one time and can be a fun lover at another time. To maintain the distinction and integrity, an adaptable privacy-based system is needed. The role of a person, as well as the requirements of that particular role, will be dynamically updated provided that the person meets the necessary security requirements. This section summarizes the requirements from different users' point of view and how Ubi-App supports these requirements.

Instructors' requirements

The most specific requirement for instructors is the capability of creating exams. For creating exams, the system should allow an instructor to enter as many questions as she/he wants. Currently, the system supports three different types of questions: Multiple choice, True/false, and Fill in the blank. Although this seems trivial, supporting the creation of an exam and facilitating its distribution requires consideration of many issues.

The instructors should be able to create the answer key for the questions whenever they like. The distribution of the exam should take place only when they select the designated action. The instructors should be able to specify the time limit for the students to submit their answers. The instructor's devices must be able to store the

data received in an efficient way, due to the memory limitations of most of the mobile devices. The instructors should also be able to receive feedback from their students. They should also be able to append any new questions to the existing exam. The instructor should have the capability of adding comments about on any specific question at any time. Also, system flexibility should permit the instructor to choose any question or part of question from any exam and add it to any other question of the same or another exam.

Students' requirements

Students should be able to submit their answers within a specified time frame. The system should be very reliable and secure, and provide the promised services. It should not deny students access to the network, reset in the middle of an exam, or partially download a file because of unspecified or invalid reasons. The students should also be able to submit their comments to the instructor during or after the exam. The students should be able to skip any question if they want. Also, they can answer any skipped question later, if time permits. Students should have note-taking capability similar to regular paper-based exams. A student should only be able to change role (e.g., fun lover) in two specific cases: the examination time has expired or an exam is finished and the answers are sent to the instructor. During an exam, a student will never be allowed to communicate with any other user except the instructor. The requirements of instructors and students are described in detail in [2].

Pollsters' requirements

The requirements of a pollster are similar to the requirements of an instructor.

Survey takers' requirements

The requirements of a survey taker are similar to those of a student, with some added requirements. No one can force any Ubi-App user to participate in a survey. Participation will only be possible if the user agrees. The survey taker can respond to all questions, or any part of a question, and can cancel the survey at any time.

Conference attendees' requirements

A conference attendee can communicate with any other attendee (e.g., to share their views about any topic, to invite another to join a specific session, etc.). A conference attendee can post any comment/question for the presenter at any time, without interrupting the session. The presenter (another conference attendee) can prioritize the comments/questions according to their importance (set by the presenter). If the presenter agrees, a conference attendee can download the presentation slides or any other necessary documents residing in the presenter's mobile device. A conference attendee can take any notes while using Ubi-App. Finally, a conference attendee can update those notes, edit them if necessary, send them to other people, etc.

Fun lovers' requirements

A fun lover should be able to maintain a list of buddies in Ubi-App, and access those who are in the vicinity using Ubi-App. Communication with buddies should take place by chatting, in a way similar to existing chat software. The Ubi-App should also maintain a list of anonymous but trusted (having intention to communicate with others) fun lovers that reside in the Wi-Fi region. Anonymous people can also be included in one's buddy list with mutual understanding/agreement. Besides chat, one can send any file (text/binary) to another and can receive any file as well. A fun lover can also simultaneously access other important applications such as notepad, image viewer, etc., if necessary.

Resident doctors' requirements

A customized form should be maintained in the Ubi-App, so that a resident doctor can easily access all necessary information about a patient. The doctor can update any additional information. A doctor can easily send all the information (or some selected information) to as many people as needed. Except for the designated authenticated doctor, no one can access the database containing patient information.

Development of Ubi-App using Ubicomp Assistant

The architecture of the MARKS middleware is shown in Fig. 1.

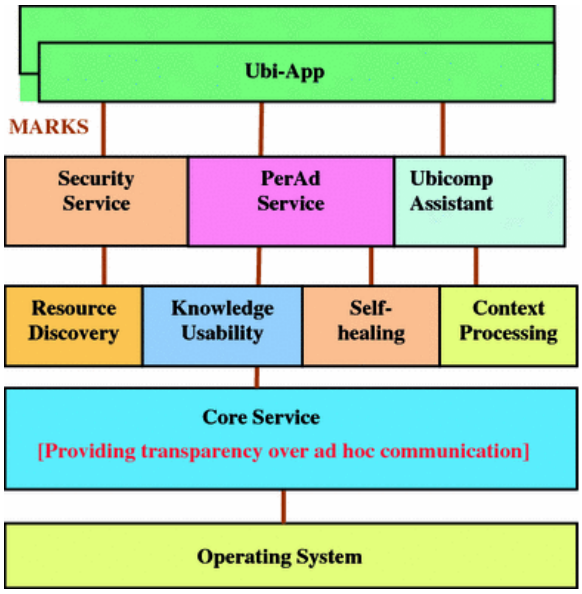

Fig. 1 MARKS architecture [7]

Ubicomp Assistant is a third tier service of MARKS

MARKS provides three-tiered services:

1. First tier services: These services are also known as core services. They ensure the transparency of ad hoc communication.
2. Second tier services: These may use the first tier services as a component, and include Resource Discovery, Knowledge Usability, Self-healing, and Context Processing services. Each service incorporates an agent.
3. Third tier services: These may use both the first tier and second tier services as components. This tier contains the Security Service, PerAd Service, and UA.

Interface definition language (IDL) and Classifier are two other basic components, which sit on top of these three tiers and maintain the communication between the application developer and all three types of services.

Ubicomp Assistant is an example of a third tier service. After converting and classifying a request from the application developer, the IDL and the classifier of MARKS send the request to the appropriate agent. This agent drives it to the appropriate unit. For example, if an application developer wants to find the nearby people who are using Ubi-App, the UA will be invoked through IDL by using Ubi-App. Then the classifier will determine the appropriate agent (here Resource Discovery agent) with the specific parameters. The Resource Discovery unit will then perform those functions by using core services and will return the result to UA via the Resource Discovery agent. Finally, UA will propagate the same result to the application developer. The entire process is transparent to both the application developer and end user.

How Ubi-App adheres to all the characteristics

In Sect. 3 the essential characteristics of Ubi-App were mentioned. Some characteristics are maintained by the UA service itself, while others are upheld by the first tier services. Heterogeneous platform flexibility (C3), ad hoc wireless infrastructure (C5), and persistent connectivity (C8) are supported by the core services of MARKS.

Ubi-App itself provides flexibility of customization (C1). Currently, Ubi-App supports user roles, such as instructors, students, pollsters, fun lovers, conference attendees, and resident doctors. The features and the interface are changed accordingly when the specific user role is chosen. Also, different types of utility functions are maintained. For example, a conference attendee will see the interface only for chatting, file transfer, and note-taking facility; while the interface for the doctor will include a patient database, as well as new patient add, edit, search, and delete facilities. A resident doctor will also be able to communicate with other nearby resident doctors through the transfer and chatting facilities.

Ubi-App supports multi dimensional utility functions (C2) like chatting with nearby interested people, secured various data (text, video, audio) transfer, assessment on the fly, survey on the fly, patient tracking, note taking by using both PDA keyboard and transcriber. It also maintains different types of databases for different types of users. For example, the database for a resident doctor would be different from the database for an instructor or pollster.

Ubi-App also satisfies the requirements of omnipresent seamless service provision (C4). It can be used at anyplace and at anytime, provided that the device on which Ubi-App is running has wireless capability. To prevent interruptions or inconveniences during the entire process, Ubi-App calls on assistance from the first and second tier services. The memory footprint (C6) for Ubi-App is very simple. Table 1 illustrates it.

Table 1 Miniature footprint of Ubi-App

	Line of code	Executable file size (KB)
MARKS-ORB	868	16
Ubi-App		
Teacher	4,586	236
Student	1,487	
Pollster	3,594	
Resident doctor	1,819	
Fun lover	96	
Conference attendee	453	
Control panel	1,128	
Authentication	211	
Others	1,182	

It is planned to make Ubi-App and the MARKS middleware freely available (C7) to all interested users via the Internet. The users will be able to easily install these in their mobile devices.

Ubi-App has the capability of transparently coupling the physical world to the mobile information world (C8). It provides the user with a variety of services, such as finding friends, chatting with them, note taking during a meeting or a class, or sharing different types of files (movies, pictures, or music files), etc. UA's services are more extensive than those offered by regular mobile devices whose services are limited or not available at all.

Use of Ubi-App

To access the application, a user requires the following:

1. a handheld device with Windows CE operating system,
2. MARKS to be deployed in that device,
3. a wireless card (IEEE 802.11b or Bluetooth).

To use the application, the user needs to follow the subsequent steps:

Step 1: Establish a wireless connection to the desired wireless network

First, the user needs to make sure that wireless communication is online. In general, to communicate successfully with other devices, each device needs to be connected in a common network. After being authenticated (if needed), the device can communicate with all other devices in that same network.

Step 2: Run MARKS

Second, after being connected to the desired wireless network, the user needs to activate MARKS, which is run in the backend. MARKS will act as the bridge between the application and Windows CE.

Step 3: Run the application

Finally, the user needs to run Ubi-App, which automatically provides the user with the necessary services.

Implementation and evaluation

Following the implementation of a prototype of Ubi-App, a twofold evaluation approach was adopted.

1. User study
2. Performance measurement

Prototype implementation

To implement a prototype of Ubi-App, WinCE was used, running on a set of Dell Axim X30 pocket PCs (Processor type is Intel@PXA270, speed is 624 MHz, display is 3.5” Transflective TFT colour, and weight is 4.8 oz). This application is also compatible with laptops, desktops, and smart phones. C# in the Compact Dot Net Framework was used as an implementation language. The mobile ad hoc mode of IEEE 802.11b, was used as the underlying wireless protocol. Experiments were also conducted with Bluetooth, obtaining similar performance. Socket and thread programming were used to make the entire communication scheme simple yet effective. Databases were maintained through SQLCE. Screen shots of the prototype are shown in Fig. 2.

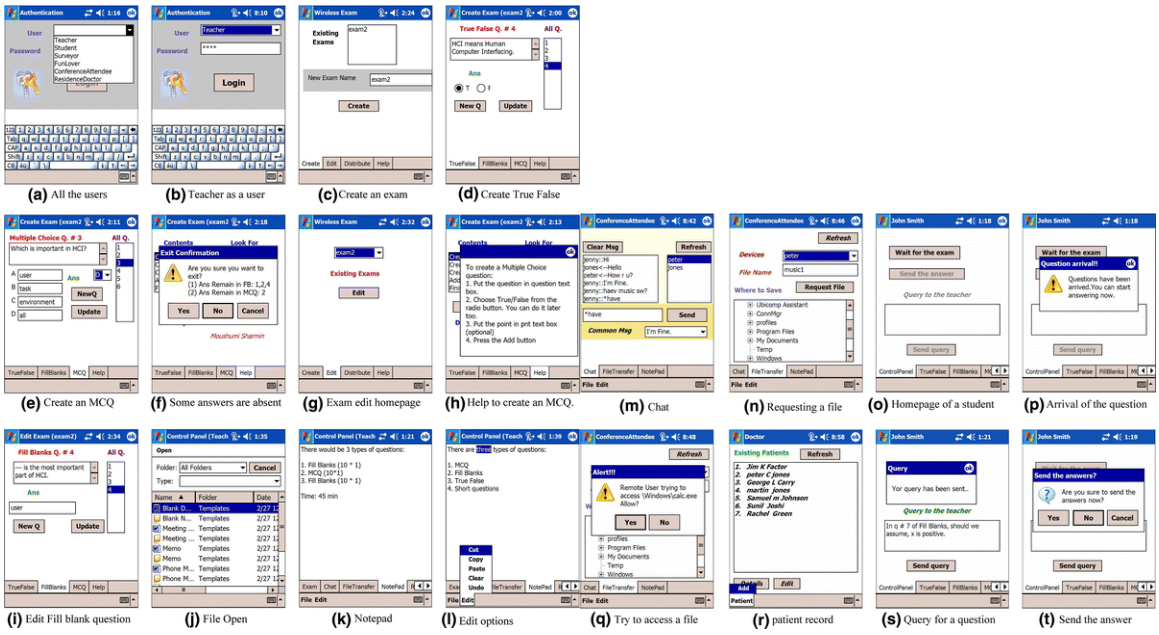

Fig. 2 Screen shots of Ubi-App

User-based evaluation

To evaluate the usability and performance of the prototype a user study was conducted with prospective users. It was planned to follow a cognitive Walkthrough Strategy [5] as this allows a single evaluator or a group of evaluators to inspect a user interface by walking through a set of tasks and assessing its understandability and ease of learning. However, at a later stage, participants from different groups were involved (not only experts but also a group representative of actual users). The experimental setup is described below (Table 2).

Table 2 Evaluation of Ubi-App

Personnel needed for the evaluation	Usability issues covered
Users: at least one type of user from each group	Effectiveness: YES
	Efficiency: YES
	Satisfaction: YES
Can be conducted remotely: NO	Can obtain quantitative data: YES

Users

A heterogeneous group of end users was involved in the experiment (17 users, 5 females, average age 36), in order to obtain practical as well as effective feedback. The group included three computer scientists, one graduate student of mechanical engineering, one undergraduate student, one college instructor, and nine professors (two from computer science, one from communication science, and six from chemistry) who attend conference regularly, as well as one English teacher, and one physician who has the experience of being a resident doctor.

Tasks

Some tasks are specific for certain user roles, such as maintaining a patient database, which is only applicable for physicians. Other tasks may be common to similar user roles, such as transferring files, which can be used by both a fun lover and a conference attendee. The tasks to be analyzed were selected in such a way that no major task has been overlooked. Before each task, the subtasks sequence leading to desired results was briefly explained to the users. It was recommended to use the “HELP” section whenever in need of assistance.

Process

A two-stage evaluation process was followed for Ubi-App. In the initial stage, a first user study was conducted based on the implemented prototype. A questionnaire was then distributed to the users to complete (see “Appendix”). After reviewing the users’ informative responses, the interface was modified design and a second prototype was implemented.

Figure 3 shows the result of the questionnaire (overall rating, ease of use, ease of input, ease of navigation, and help section). Figure 4 shows the results by age, and Fig. 5 by technical background. These three figures implicitly prove the scalability of Ubi-App, since the overall average rating is more than four irrespective of the users’ age and technical background. Ubi-App has also been rated from the perspective of the application developer. Figure 6 shows the results.

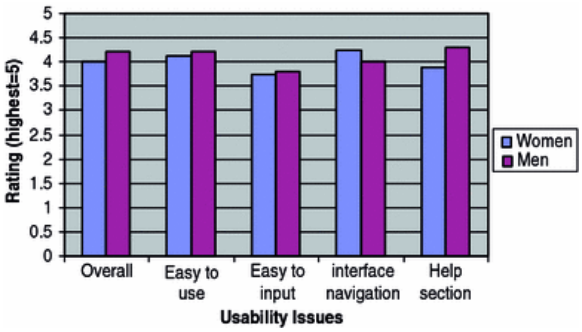

Fig. 3 Rating by users based on gender

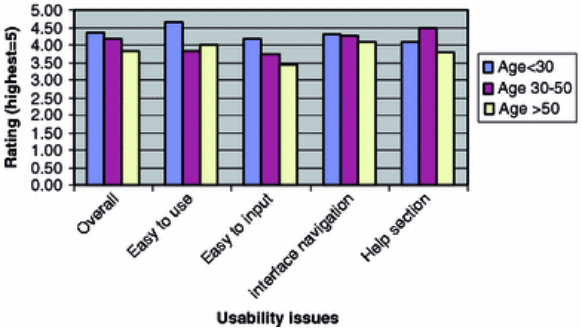

Fig. 4 Rating by users based on age

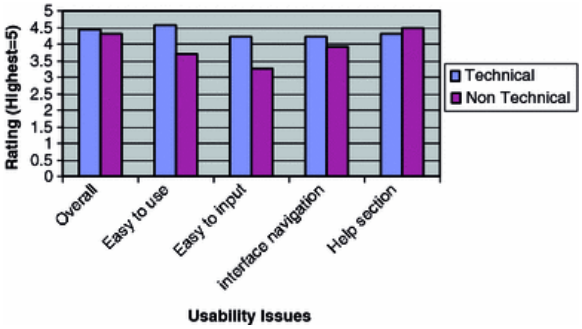

Fig. 5 Rating by users based on technical background

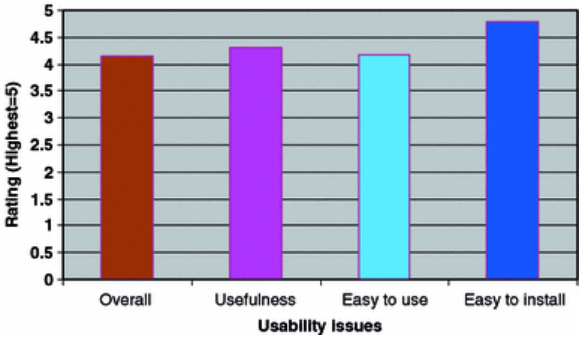

Fig. 6 Rating by application developer

Performance measurement

Ubi-App does not have any adverse affect on the performance of handheld devices. The main two performance metrics are power consumption and signal strength. It was observed that Ubi-App consumes little power. Figure 7 shows the power consumption of four PDAs while the running application is Ubi-App. To make the comparison crystal clear, two cases are shown for each PDA: an idle case (with the PDA “on” but not doing anything) and a run case (when the application is running on the PDA). Performance measurement based on signal strength is shown in Fig. 8, with results similar to power consumption.

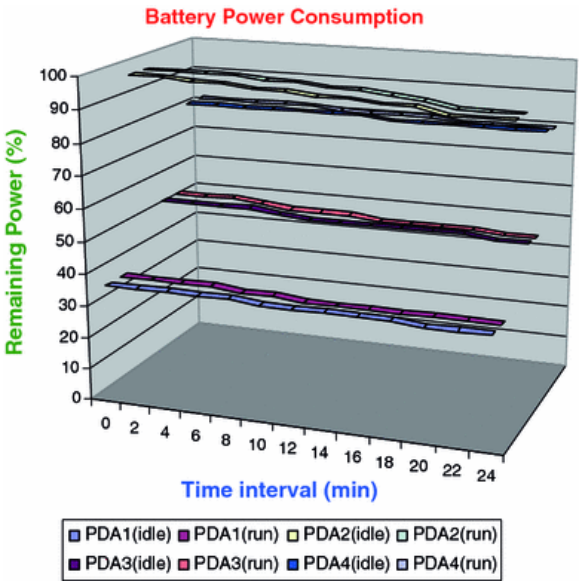

Fig. 7 Power consumption by all PDAs before and after running the Ubi-App

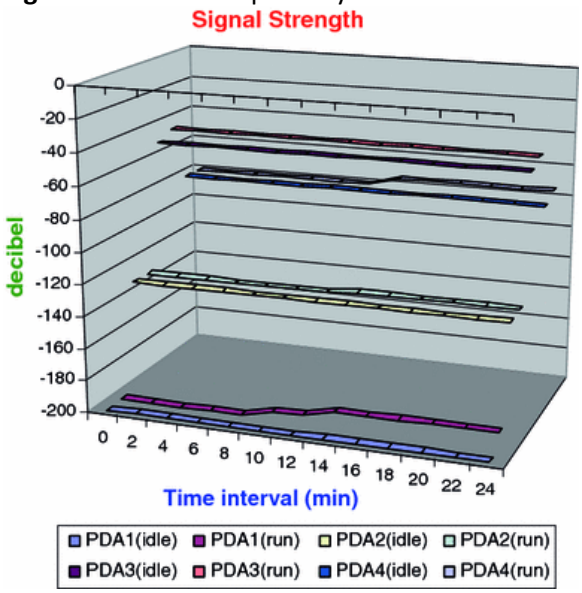

Fig. 8 Signal strength by all PDAs before and after running the Ubi-App

Conclusions and future work

An important emerging requirement in ubiquitous computing is the availability of omnipresent customizable middleware services, which can be utilized by different users. To accomplish this, UA was developed, a three-tier middleware service of MARKS. Moreover, to show the efficiency of this service, an application, called Ubi-App, was implemented, that uses services which include, but are not limited to, wireless ad hoc chatting, wireless file transfer using pervasive technology, note-taking capability, all-in-one exam package, effective survey provisions, secured sharing patient record, etc. This application uses IEEE 802.11 for free short-range wireless networking. This mode makes Ubi-App available anywhere, anytime.

This paper has described the details of the Ubi-App application along with its characteristics and functional requirements. This application was evaluated with the involvement of representatives from different end user groups. Their comments were carefully considered and are presented in the result section. After receiving feedback from the end users, the related changes were incorporated in the prototype. Although most of the

suggestions of the users were taken into account, there are still areas for further improvement. One area of future work includes the incorporation of voice commands. The users commented that operating a PDA would be lot easier if it can take inputs as simple voice commands. Another research direction could also include user verification by means of fingerprint and/or signature, since privacy and security are two main issues for many applications. This feature is currently being implemented.

References

1. 100+ Applications for PDA. http://www.amazon.com/o/ASIN/B00006ISHC/002-4257409-5636800?SubscriptionId=1NP56Y8QKPN1AZERGY02. Accessed May 6, 2007
2. Ahamed, S., Sharmin, M., Ahmed, S., Havice, M., Anamanamuri, S.: An assessment tool for out of class learning using pervasive computing technologies. Inf. J. **8**(5), 71–767 (2005)
3. Ahn, J., Pierce, J.: SEREFE: serendipitous file exchange between users and devices. Mobile HCI 2005 (2005)
4. Bernstein, P.: Middleware: a model for distributed services. Commun. ACM **39**(2) (1996)
5. Cognitive Walkthrough Strategy. http://www.pages.drexel.edu/~zwz22/CognWalk.htm. Accessed May 6, 2007
6. Dey, A., Salber, D., Abowd, G., Futakawa, M.: The conference assistant: combining context-awareness with wearable computing. In: International Semantic Web Conference (1999)
7. Fischer, G., Konomi, S.: Innovative media in support of distributed media and lifelong learning. In: WMTE 2005 (2005)
8. Gaia: Active Space for Ubiquitous Computing. http://gaia.cs.uiuc.edu/html/projects.htm. Accessed May 6, 2007
9. Geihs, K.: Middleware challenges ahead. IEEE Comput. J. **34**(6) (2001)
10. Holzinger, A.: Application of rapid prototyping to the user interface development for a virtual medical campus. IEEE Softw. **21**(1), 92–99. ISSN: 0740-7459 (2004)
11. Holzinger, A.: Usability engineering for software developers. Commun. ACM **48**(1), 71–74. ISSN: 0001-0782 (2005)
12. Holzinger, A., Errath, M.: Designing web-applications for mobile computers: experiences with applications to medicine. In: Stephanidis, C.; Stary, Chr. (eds.) User Centered Interaction Paradigms for Universal Access in the Information Society. Lecture Notes in Computer Science, LNCS, vol. 3196. ISBN: 3-540-23375-X (2004)
13. Karagiannis, G., Stamatopoulos, V., Roussos. G.: Assessing patient case management services with ACHD sensed data. In: Second International Workshop on Body Sensor Networks (2005)
14. MIT Project Oxygen: Pervasive human-centered computing, Project Overview, July 2005. http://oxygen.lcs.mit.edu/Overview.html. Accessed May 6, 2007
15. Ranganathan, A., Campbell, R., Ravi, A., Mahajan, A.: ConChat: a context-aware chat program. IEEE Pervasive Comput. (2002)
16. Sharmin, M., Ahmed, S., Ahamed, S.: MARKS (middleware adaptability for resource discovery, knowledge usability and self-healing) in pervasive computing environments. In: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG 2006), Las Vegas, Nevada, USA, pp 306–313 (2006)
17. Sharmin, M., Ahmed, S., Ahamed, S.: Ubicomp assistant: an omnipresent customizable service using MARKS. In: Proceedings of the 21st Annual ACM Symposium on Applied Computing (ACM SAC 2006), Dijon, France, pp 1013–1017 (2006)
18. Statistics for mobile commerce. http://www.epaynews.com/statistics/mcommstats.html#7. Accessed May 6, 2007
19. Stephanidis, C.: Adaptive techniques for universal access. User Model. User-Adapt. Interact. **11**(1–2), 159–179 (2001)
20. Stephanidis, C., Savidis, A.: Universal access in the information society: methods, tools, and interaction technologies. Universal Access Inf. Soc. **1**(1), 40–55 (2001)
21. Stewart, D.: Miniature software for large pervasive computing applications. In: Invited position paper at New Visions for Software Design Productivity Workshop (2001)

22. Yau, S., Karim, F., Wang, Y., Wang, B., Gupta, S.: Reconfigurable context-sensitive middleware for pervasive computing. In: IEEE Pervasive Computing, joint special issue with IEEE Personal Communications, vol. 1(3) (2002)

Acknowledgments

The authors thank Paula Stroud, Karl Stamm, and members of Ubicomp Lab, Marquette University, for commenting on earlier drafts and for their valuable assistance in improving the readability of the paper. We also thank the users who participated in our study.

Appendix: Questionnaire for evaluation

Appendix: Questionnaire for evaluation

1. Overall, how would you rate the service? (1 = very bad, 5 = excellent)
2. Overall, how would you rate this application? (1 = very bad, 5 = excellent)
3. Assess the usefulness of the services: (1 = burden, 5 = very useful)
4. How easy is the application to use? (1 = very hard, 5 = very easy)
5. How easy is it for you to provide input? (1 = very hard, 5 = very easy)
6. How easy is it to navigate the interface? (1 = very hard, 5 = very easy)
7. How easy is it to establish the initial wireless connection? (1 = very hard, 5 = very easy)
8. How easy is it to get the desired output after performing an action? (1 = very hard, 5 = very easy)
9. How effective is the interface for the small screen? (1 = not at all, 5 = very effective)
10. How effective is the “Help” section? (1 = not at all, 5 = very effective)
11. Which feature do you feel is unnecessary?
12. Which feature do you think is most useful?
13. Who will benefit the most from this application?
14. Will you choose this application if it is free? If not, why?
15. Would you recommend this application to a friend? If not, why?
16. Can you recommend any other users who can use this software?
17. Can you recommend any other features that should be incorporated in this software?
18. Please indicate any other suggestions or comments for improvement that you may have.